CLAIMS

We claim:

1. 1. A method in a computer system for executing applications to request
2. services from remote shared service providers each able to provide services to multiple
3. executing applications, each remote shared service provider having an associated distinct
4. proxy component for communicating with the remote shared service provider, comprising:
5. under control of a first of the executing applications,
6. instantiating a local messaging component within the first application, the
7. local messaging component able to communicate with each of the associated proxies;
8. notifying the local messaging component to request a specified service
9. from a remote shared service provider identified with a unique name; and
10. under control of the local messaging component, requesting the specified
11. service by,
12. determining based on the unique name identified in the notification
13. to request the specified service from a first of the remote shared service providers;
14. retrieving configuration information for the first remote shared
15. service provider that identifies a first proxy component associated with the first remote
16. shared service provider;
17. when the first proxy component is not already instantiated within the
18. first application, instantiating the first proxy component within the first application;
19. sending a message to the first proxy component requesting that the
20. first remote shared service provider provide the specified service; and
21. receiving a response from the first remote shared service provider via
22. the first proxy component, and notifying the first application of the received response,
23. so that an application can communicate with any of the remote shared service providers via a
24. local messaging component that interacts with a proxy component associated with the remote
25. shared service provider.

1   2.   The method of claim 1 including, under control of the first application:

2   notifying the local messaging component to request a second specified

3 service from a remote shared service provider identified with a second unique name; and

4   under control of the local messaging component, requesting the second

5 specified service by,

6   determining based on the second unique name identified in the

7 notification to request the second specified service from a second of the remote shared

8 service providers;

9   retrieving configuration information for the second remote shared

10 service provider that identifies a second proxy component associated with the second remote

11 shared service provider;

12   sending a message to the second proxy component requesting that

13 the second remote shared service provider provide the second specified service; and

14   receiving a response from the second remote shared service provider

15 via the second proxy component, and notifying the first application of the received response.

1   3.   The method of claim 1 including:

2 under control of a second of the executing applications,

3   instantiating a second local messaging component within the second

4 application, the second local messaging component able to communicate with each of the

5 associated proxies;

6   notifying the second local messaging component to request a specified

7 service from a remote shared service provider identified with the unique name of the first

8 shared service provider; and

9   under control of the second local messaging component, requesting the

10 specified service by,

11   determining based on the unique name identified in the notification

12 to request the specified service from the first remote shared service provider;

13   retrieving the configuration information for the first remote shared

14 service provider that identifies the associated first proxy component;

15   when the first proxy component is not already instantiated within the

16 second application, instantiating the first proxy component within the second application;

17 and

18   sending a message to the first proxy component requesting that the

19 first remote shared service provider provide the specified service.

1   4.   The method of claim 1 including, under control of the first proxy

2 component:

3   receiving the sent message from the local messaging component;

4   identifying based on the received message one of multiple services provided by

5 the first remote shared service provider that is requested;

6   requesting the first remote shared service provider to provide the identified

7 service;

8   receiving a response from the first remote shared service provider; and

9   sending the received response to the local messaging component.

1   5.   The method of claim 4 including, under control of the first remote

2 shared service provider:

3   receiving the request from the first proxy component to provide the identified

4 service;

5   performing processing associated with the identified service, the processing

6 generating a response; and

7   sending the generated response to the first proxy component.

1   6.   The method of claim 1 wherein the first remote shared service provider

2 has multiple associated proxy components each for communicating with the first remote

3 shared service provider to request distinct services provided by the first remote shared

4 service provider, wherein the notifying of the local messaging component further identifies a

5   name of the specified service, and wherein the identifying of the first proxy component from

6   the retrieved configuration information is based at least in part on the identified name of the

7   specified service.

1   7.   The method of claim 1 wherein the local messaging component and the

2   first proxy component are software modules, and wherein the instantiating of the components

3   within the first application includes executing the software modules.

1   8.   The method of claim 1 wherein the local messaging component and the

2   first proxy component are instances of object classes, and wherein the instantiating of the

3   components within the first application includes creating the instances.

1   9.   The method of claim 8 wherein the notifying of the local messaging

2   component includes invoking an interface method of the local messaging component

3   instance.

1   10.   The method of claim 8 wherein the sending of the message to the first

2   proxy component includes invoking an interface method of the first proxy component

3   instance.

1   11.   The method of claim 1 wherein the configuration information is

2   retrieved from storage via a directory service using the unique name of the first remote

3   shared service provider.

1   12.   The method of claim 11 including, before the executing of the first

2   application, storing the configuration information using the directory service.

1   13.   The method of claim 1 wherein the retrieved configuration information

2   includes initialization information for the first proxy component, and wherein the

3   instantiating of the first proxy component includes providing the initialization information to

4   the first proxy component for use as part of its initialization.

1   14.   The method of claim 1 wherein the notifying of the request of the

2   specified service includes a message to be sent to the first remote shared service provider,

3   and wherein the message sent to the first proxy component is the included message.

1   15.   The method of claim 1 wherein the notifying of the request of the

2   specified service includes a message to be sent to the first remote shared service provider,

3   and wherein the message sent to the first proxy component is a version of the included

4   message that is transformed under the control of the local messaging component before the

5   sending.

1   16.   The method of claim 1 wherein the first application includes multiple

2   action handler components each for responding to requests from external clients with

3   responses generated in a format used by the action handler component, and wherein the

4   notifying of the request of the specified service is prompted by a request generated by one of

5   the action handler components.

1   17.   The method of claim 16 wherein the first application includes multiple

2   view handler components each for transforming generated responses in formats used by

3   action handler components into formats compatible with external clients, and including, after

4   the notifying of the received response from the first remote shared service provider, using

5   one of the view handler components to transform the received response into a format

6   compatible with an external client.

1   18.   A computer-readable medium whose contents cause a computing device

2   to execute applications able to request services from remote shared service providers each

3   able to provide services to multiple executing applications, each remote shared service

4   provider having an associated proxy component for communicating with the remote shared

5   service provider, by:

6   under control of a first of the executing applications,

7   instantiating a local messaging component, the local messaging

8   component able to communicate with each of the associated proxies;

9   notifying the local messaging component to request a specified service

10   from a remote shared service provider; and

11   under control of the local messaging component, requesting the specified

12   service by,

13   determining based on the notifying to request the specified service

14   from a first of the remote shared service providers;

15   retrieving configuration information for the first remote shared

16   service provider that identifies a first proxy component associated with the first remote

17   shared service provider;

18   when the first proxy component is not already instantiated,

19   instantiating the first proxy component;

20   sending a message to the first proxy component requesting that the

21   first remote shared service provider provide the specified service; and

22   receiving a response from the first remote shared service provider via

23   the first proxy component.

1   19.   A computing device for executing applications to request services from

2   remote shared service providers each able to provide services to multiple executing

3   applications, each remote shared service provider having an associated proxy component for

4   communicating with the remote shared service provider, comprising:

5   a first application capable of, during execution, instantiating a local messaging

6   component able to communicate with each of the associated proxies and of notifying the

7   local messaging component to request a specified service from a remote shared service

8   provider; and

9   the local messaging component capable of, during execution, determining

10   based on the notification to request the specified service from a first of the remote shared

11   service providers, of retrieving configuration information for the first remote shared service

12   provider that identifies a first proxy component associated with the first remote shared

13   service provider, of instantiating the first proxy component, of sending a message to the first

14   proxy component requesting that the first remote shared service provider provide the

15   specified service, and of receiving a response from the first remote shared service provider

16   via the first proxy component.

1   20.   A method in a computer system for an executing application to

2   communicate with remote services via a local messaging service, each of the remote services

3   having an associated proxy for communicating with the remote service, the method

4   comprising:

5   for each of multiple of the remote services,

6   notifying the local messaging service to send a message to the remote

7   service;

8   under control of the local messaging service,

9   identifying the proxy associated with the remote service by

10   retrieving information associated with the remote service; and

11   sending the message to the identified proxy for communication to the

12   remote service; and

13   receiving a response to the sent message from the remote service via the

14   identified proxy.

1   21.   The method of claim 20 including, under control of each of the

2   identified proxies:

3   receiving the sent message from the local messaging service that is to be

4   communicated to a remote service;

5   sending the received message to the remote service;

6   receiving a response from the remote service; and

7   sending the received response to the local messaging service.

1   22. The method of claim 21 including, before the sending of the received

2 message to the remote service, transforming the received message into a format appropriate

3 for the remote service.

1   23. The method of claim 20 wherein each of the remote services has

2 multiple associated proxies, wherein the notifying of the local messaging service identifies a

3 manner of sending the message to the remote service, and wherein the identifying of the

4 associated proxy from the retrieved information includes selecting one of the multiple

5 associated proxies based at least in part on the identified manner.

1   24. The method of claim 20 wherein the local messaging service and the

2 proxies are created objects, and including creating the objects.

1   25. The method of claim 20 wherein the information associated with a

2 remote service is retrieved from a directory service using a unique name of the remote

3 service.

1   26. The method of claim 25 including storing the associated information

2 with the directory service before the retrieving.

1   27. The method of claim 20 wherein the application includes multiple

2 action handlers each able to respond to requests from external clients with responses

3 generated in a format used by the action handler, and wherein the notifying of the local

4 messaging service to send the messages is prompted by requests generated by the action

5 handlers.

1   28. The method of claim 27 wherein the application includes multiple view

2 handlers each able to transform a generated response in a format used by an action handler

3   into a format compatible with at least one of the external clients, and including, after the

4   receiving of the response from the remote service, using one of the view handlers to

5   transform the received response into a format compatible with an external client.

1   29.   A computer-readable medium whose contents cause a computing device

2   to execute an application able to communicate with remote services via a local messaging

3   service, each of the remote services having an associated proxy for communicating with the

4   remote service, by:

5   for each of multiple of the remote services,

6   notifying the local messaging service to send a message to the remote

7   service;

8   under control of the local messaging service,

9   identifying the proxy associated with the remote service by

10   retrieving information associated with the remote service; and

11   sending the message to the identified proxy for communication to the

12   remote service; and

13   receiving a response to the sent message from the remote service via the

14   identified proxy.

1   30.   The computer-readable medium of claim 29 wherein the computer-

2   readable medium is a data transmission medium transmitting a generated data signal

3   containing the contents.

1   31.   The computer-readable medium of claim 29 wherein the computer-

2   readable medium is a memory of a computer system.

1   32.   A computing device for executing an application to communicate with

2   remote services via a local messaging service, each of the remote services having an

3   associated proxy for communicating with the remote service, comprising:

4   an application capable of, for each of multiple of the remote services, notifying

5   the local messaging service to send a message to the remote service and receiving a response

6   to the sent message from the remote service via a proxy associated with the remote service;

7   and

8   the local messaging service capable of identifying the proxy associated with the

9   remote service by retrieving information associated with the remote service and of sending

10   the message to the identified proxy for communication to the remote service.

1   33.   A computer system for executing an application to communicate with

2   remote services via a local messaging service, each of the remote services having an

3   associated proxy for communicating with the remote service, comprising:

4   means for, for each of multiple of the remote services, notifying the local

5   messaging service to send a message to the remote service and receiving a response to the

6   sent message from the remote service via a proxy associated with the remote service; and

7   means for identifying the proxy associated with the remote service by

8   retrieving information associated with the remote service and sending the message to the

9   identified proxy for communication to the remote service.

1   34.   A method in a computer system for invoking functions of remote

2   applications, each of the remote applications having an associated proxy for communicating

3   with the remote application, the method comprising:

4   for each of multiple of the remote applications,

5   receiving a request to invoke a specified function of the remote

6   application using at least one specified parameter value;

7   retrieving information associated with the remote application that

8   identifies the proxy associated with the remote application;

9   when a copy of the identified proxy has not already been instantiated,

10   instantiating a copy of the identified proxy;

11   notifying the identified proxy copy of the specified function and the

12   specified parameter values;

13   under control of the identified proxy copy, invoking the specified function

14   of the remote application using the specified parameter values; and

15   receiving a response from the invoking of the specified function.

1   35.   The method of claim 34 wherein each of the remote applications has

2   multiple associated proxies for invoking distinct functions of the remote application, and

3   wherein the identifying of the proxy associated with the remote application from the

4   retrieved information is based at least in part on the specified function.

1   36.   The method of claim 34 wherein the proxies are object instances, and

2   wherein the instantiating of the proxies includes creating the object instances.

1   37.   The method of claim 34 wherein the information is retrieved from a

2   directory service using an identified name of the remote application.

1   38.   The method of claim 34 wherein the retrieved information includes

2   initialization information for the identified proxy, and wherein the instantiating of the proxy

3   includes providing the initialization information to the proxy for use during initialization.

1   39.   The method of claim 34 wherein the application includes multiple

2   action handlers each able to respond to requests from external clients with responses

3   generated in a format used by the action handler, and wherein the request to invoke the

4   specified function is received from one of the action handler components.

1   40.   The method of claim 39 wherein the application includes multiple view

2   handlers each able to transform responses in formats used by action handlers into formats

3   compatible with external clients, and including, after the receiving of the response from the

4   invoking of the specified function, using one of the view handler components to transform

5   the received response into a format compatible with an external client.

1   41.   A computer-readable medium whose contents cause a computing device

2   to invoke functions of remote applications each having an associated proxy for

3   communicating with the remote application, by:

4   for each of multiple of the remote applications,

5   receiving a request to invoke a specified function of the remote

6   application using at least one specified parameter value;

7   retrieving information associated with the remote application that

8   identifies the proxy associated with the remote application;

9   when a copy of the identified proxy has not already been instantiated,

10   instantiating a copy of the identified proxy;

11   notifying the identified proxy copy of the specified function and the

12   specified parameter values;

13   under control of the identified proxy copy, invoking the specified function

14   of the remote application using the specified parameter values; and

15   receiving a response from the invoking of the specified function.

1   42.   A computing device for invoking functions of remote applications each

2   having an associated proxy for communicating with the remote application, comprising:

3   a first component capable of, for each of multiple of the remote applications,

4   receiving a request to invoke a specified function of the remote application using at least one

5   specified parameter value, retrieving information associated with the remote application that

6   identifies the proxy associated with the remote application, instantiating a copy of the

7   identified proxy, notifying the identified proxy copy of the specified function and the

8   specified parameter values, and receiving a response from the invoking of the specified

9   function; and

10   multiple proxy copies each capable of receiving a notification of a specified

11 function and specified parameter values for one of the remote applications and of invoking

12 the specified function of the remote application using the specified parameter values.

1   43.   A method in a computer system for components of an executing

2 application to communicate with remote services, each of the remote services having an

3 associated proxy for communicating with the remote service, the executing application

4 having a messaging service component for communicating with the associated proxies,

5 having multiple action handler components each for responding to requests with responses

6 generated in a format used by the action handler component, and having multiple view

7 handler components each for transforming generated responses in formats used by action

8 handler components into formats compatible with external clients, the method comprising:

9   under control of each of multiple of the action handler components,

10   receiving a request from a remote client to provide a service; and

11   providing the requested service by,

12   determining a need for functionality available from one of the remote

13 services;

14   notifying the local messaging component to request the needed

15 functionality from the one remote service;

16   receiving a response from the one remote service via the local

17 messing component;

18   generating a response in a format used by the action handler

19 component;

20   determining a view handler component able to transform the

21 generated response from the format used by the action handler component to a format

22 compatible with the remote client; and

23   providing the generated response to the determined view handler

24 component for the transforming;

25   under control of the local messaging component,

26   receiving notifications from multiple action handler components each to

27 request functionality from a remote service; and

28   for each received notification,

29   identifying the proxy associated with the remote service; and

30   sending a request for the functionality to the identified proxy for

31 communication to the remote service;

32   receiving a response to the sent request from the remote service via

33 the identified proxy; and

34   notifying the action handler component of the received response; and

35   under control of each of multiple of the view handler components,

36   receiving responses generated by at least one action handler component in

37 a format used by the action handler component; and

38   for each received response, transforming the received response into a

39 format that is compatible with the remote client for whom the received response was

40 generated.